REMARKS

Claims 1 - 15 remain pending in the present application.

Applicant respectfully submits that pending claims 1 - 15 are patentable over the cited art of record. Accordingly, reconsideration and allowance of the application and presently pending claims 1 - 15 are respectfully requested.

Claim Rejections Under 35 U.S.C. §103(a) - Claims 1 - 15

A. Statement of the Rejection

The Office Action indicates that claims 1 - 7, 9, 10, and 12 - 15 stand rejected under 35 U.S.C. 103(a) as allegedly being unpatentable over U.S. Patent No.: 5,787,286, to Hooker, hereafter *Hooker*, in view of U.S. Patent No. 5,644,709 to Austin, hereafter *Austin*.

The Office Action indicates that claims 8 and 11 stand rejected under 35 U.S.C. 103(a) as allegedly being unpatentable over *Hooker* and *Austin*, in view of U.S. Patent No. 5,526,421 to Houldsworth, hereafter *Houldsworth*.

B. Discussion of the Rejection

Applicant respectfully traverses the rejections of claims 1 - 7, 9, 10, and 12 - 15 and claims 8 and 11. In order for a claim to be properly rejected under 35 U.S.C. §103, "[t]he PTO has the burden under section 103 to establish a *prima facie* case of obviousness. To establish a *prima facie* case of obviousness, the prior art reference (or references when combined) must teach or suggest all the claim limitations. The teaching or suggestion to make the claimed combination and the reasonable expectation of success must both be found in the prior art and not based on applicant's disclosure. *In re Vaeck*, 947 F.2d 488, 20 USPQ2d 1438 (Fed. Cir. 1991).

In this regard, the cited references (*i.e., Hooker, Austin, and Houldsworth*) do not show the combination of elements recited in Applicant's claimed invention. Thus, the cited references fail to meet the burden of disclosing, teaching, or suggesting each feature of Applicant's claimed invention. Consequently, for at least this reason the rejection fails to establish a *prima facie* case of obviousness when applied to Applicant's claims 1 - 15. Accordingly, for at least this reason, the claim rejections under 35 U.S.C. §103 should be withdrawn.

Specifically, and with particular regard to claims 1 - 7, 9, 10, and 12 - 15, each of Applicant's independent claims include at least one element that is not disclosed, taught, or suggested by the apparatus for performing software performance checks apparently disclosed in *Hooker* and the method for detecting memory access errors apparently disclosed in *Austin*.

With particular regard to claims 8 and 11, each of these claims include at least one element that is not disclosed, taught, or suggested by the proposed combination of *Hooker*, *Austin*, and *Houldsworth*.

In addition, Applicant disagrees with the assertion that one of ordinary skill in the art would be motivated to combine *Hooker's* tabulation instructions with the teachings of *Austin*. As explained in the Background of the Invention section, *Hooker* is directed to a real-time software solution for obtaining performance data that does not occupy processor time. See *Hooker*, column 1, lines 51 - 56. As further explained in the Disclosure of the Invention section, *Hooker* apparently discloses inserting at least one tabulation instruction in a first plurality of instructions and executing the tabulation instruction using a second execution unit. By executing the tabulation instruction in a second execution unit, the system of *Hooker* can be used to monitor or otherwise measure the execution performance of the first plurality of instructions without degrading the execution performance of the first plurality of instructions. See *Hooker*, column 2, lines 5 - 24. As admitted by the Office in the statement of the rejection, *Hooker* does not specify the nature of, nor does *Hooker* suggest, inserting a correctness check function associated with a particular portion of the first set of instructions.

To overcome the admitted failure of *Hooker* to disclose, teach, or suggest Applicant's claimed invention, the statement of the rejection alleges that *Austin* in a method to add instrumentation code for performance evaluation analogous to Hooker, discloses correctness checks on memory bounds in order to improve performance. Office Action, page 3, lines 20 - 22. The statement of the rejection then alleges that one of ordinary skill in the art at the time the invention was made would add the performance check/measurement instructions of *Hooker* to the memory-related correctness checking instructions by *Austin* to avoid conflict while handling resources.

Applicant respectfully disagrees with the assertion that the proposed combination of *Hooker* and *Austin* obviates Applicant's claimed method and

apparatus. First, Applicant's claims distinguish Applicant's claimed apparatus and method from the systems of both *Hooker* and *Austin*. Second, Applicant notes that *Hooker* and *Austin* do not share a common intent. *Hooker's* motivation is to monitor run performance without degrading execution performance of the first plurality of instructions. *Austin's* intent is to verify correctness of spatial and temporal dereferences to pointers in a program using an improved debugger. In contrast with the intent of *Hooker* to not degrade execution performance of a compiled program, *Austin*, on its face, indicates that the method for converting a preexisting source-language program file into a safe program includes the insertion of memory-access error checking code into the program. *Austin* optimizes the debugger by skipping memory-access error checking code during periods when no new memory deallocations have occurred. See *Austin*, Summary of the Invention, column 7, line 64 to column 8, line 17. Consequently, one of ordinary skill in the art would not be motivated to combine an intrusive memory-access error checking code as apparently taught by *Austin* with *Hooker's* tabulation instructions, as the result of the proposed combination would negate the alleged advantages of the teachings of *Hooker*.

Hooker by all indications is unwilling to degrade execution performance. *Austin*, on the other hand, expressly teaches adding code in order to verify an operationally safe program (*i.e.*, no spatial or temporal memory- access errors). For at least the reason that the Office combines teachings that contradict one another, Applicant submits that one skilled in the art would not be motivated to combine the systems and methods of *Hooker* with those apparently described by *Austin*.

As stated above, Applicant's apparatus and methods as recited in independent claims 1, 4, 6, and 9 are not rendered obvious over the proposed combination of *Hooker* and *Austin* for at least the additional reason that the proposed combination does not teach or suggest all elements or features of Applicant's claims.

1. Claims 1 - 3

Applicant's independent claim 1 is exemplary. For convenience of analysis, independent claim 1 is repeated on the following page in its entirety.

> 1. An apparatus for performing correctness checks, the apparatus comprising:
>
> logic configured to receive a first set of instructions;
>
> ***logic configured to generate an initial instruction schedule and a conditional instruction stream from the first set of instructions***, such that the initial instruction schedule is devoid of code sequences comprising correctness check functions and such that code sequences of the conditional instruction stream are associated with a corresponding set of one or more instructions in the initial instruction schedule;
>
> logic configured to evaluate the initial instruction schedule to determine whether the initial instruction schedule includes spare instruction slots into which said code sequences associated with correctness check functions can be inserted into the initial instruction schedule such that a final instruction schedule responsive to the initial instruction schedule would not require a longer run time than the initial instruction schedule; and
>
> ***logic configured to generate the final instruction schedule responsive to the initial instruction schedule, the conditional instruction stream, and the logic configured to evaluate.***

(Applicant's independent Claim 1 - *emphasis added.*)

The cited art of record fails to disclose, teach, or suggest at least the emphasized elements of pending claim 1 as shown above. Consequently, claim 1 is allowable.

Specifically, the systems disclosed in *Hooker* and *Austin* fail to disclose, teach, or suggest Applicant's claimed "***logic configured to generate an initial instruction schedule and a conditional instruction stream from the first set of instructions . . .***" Both *Hooker* and *Austin* are silent regarding logic configured to generate an initial schedule and a conditional instruction stream from the first set of instructions.

In this regard, the Office alleges that FIG. 1 of *Hooker i.e.*, "locating of bubbles by a compiler implicitly discloses analyzing of program scheduler data," teaches Applicant's claimed logic configured to generate an initial instruction schedule and an instrumentation instruction stream from a first set of instructions.

FIG. 1 of *Hooker* does not disclose, teach, or suggest the generation of an initial instruction schedule and a conditional instruction stream from a first set of instructions. The related detailed description specifically describes a list of floating point operations into which integer operations are inserted at the location of a bubble.

Conversely, *Hooker* appears to suggest that floating point operations could be inserted into a list of integer operations at an integer bubble. *Hooker's* bubble locating does not disclose, teach, or suggest the generation of an initial instruction schedule and an instrumentation instruction stream from a first set of instructions. *Austin's* improved debugging method for ensuring a memory-access safe program does not remedy the failure of *Hooker* to disclose, teach, or suggest the creation of an initial instruction schedule and an instrumentation instruction stream from a first set of instructions.

Moreover, because the systems disclosed in *Hooker* and *Austin* fail to disclose, teach, or suggest Applicant's claimed logic that generates an initial instruction schedule and an instrumentation instruction stream, both *Hooker* and *Austin* cannot disclose, teach, or suggest Applicant's claimed "***logic configured to generate the final instruction schedule responsive to the initial instruction schedule, the conditional instruction stream, and the logic configured to evaluate***." Thus, claim 1 is allowable over the proposed combination and the rejection should be withdrawn.

Because independent claim 1 is allowable, dependent claims 2, 3, and 12 - 14 are also allowable. *See In re Fine*, 837, F.2d 1071, 5 U.S.P.Q.2d 1596, 1598. (Fed. Cir. 1988.) Accordingly, Applicant respectfully requests that the rejection of claims 1 - 3 and 12 - 14 be withdrawn.

2. Claims 4, 5, and 15

For convenience of analysis, independent claim 4 is repeated below in its entirety.

> 4. An apparatus for performing correctness checks, the apparatus comprising:
>
> means for receiving a first set of instructions;
>
> ***means for generating an initial instruction schedule and a conditional instruction stream from the first set of instructions***, such that the initial instruction schedule is devoid of code sequences comprising correctness check functions and such that code sequences of the conditional instruction stream are associated with a corresponding set of one or more instructions in the initial instruction schedule;
>
> means for evaluating the initial instruction schedule to determine whether the initial instruction schedule includes spare instruction slots into which said code sequences associated with correctness check functions can

> be inserted into the initial instruction schedule such that a final instruction schedule responsive to the initial instruction schedule would not require a longer run time than the initial instruction schedule; and
>
> means for inserting said code sequences associated with the correctness check function into the spare instruction slots if enough spare instruction slots exist in the initial instruction schedule for accommodating said code sequences.

(Applicant's independent Claim 4 - *emphasis added.*)

The cited art of record fails to disclose, teach, or suggest at least the emphasized elements of pending claim 4 as shown above. Consequently, claim 4 is allowable.

Specifically, the systems disclosed in *Hooker* and *Austin* fail to disclose, teach, or suggest Applicant's claimed "***means for generating an initial instruction schedule and a conditional instruction stream from the first set of instructions . . .***" Both *Hooker* and *Austin* are silent regarding means for generating an initial schedule and a conditional instruction stream from the first set of instructions.

Hooker's bubble locating does not disclose, teach, or suggest means for generating an initial instruction schedule and an instrumentation instruction stream from a first set of instructions. *Austin's* improved debugging method for ensuring a memory-access safe program does not remedy the failure of *Hooker* to disclose, teach, or suggest Applicant's claimed means for generating an initial instruction schedule and an instrumentation instruction stream from a first set of instructions.

Because independent claim 4 is allowable, dependent claims 5 and 15 are also allowable. *See In re Fine, supra.* Accordingly, Applicant respectfully requests that the rejection of claims 4, 5, and 15 be withdrawn.

3. Claims 6 and 7

For convenience of analysis, independent claim 6 is repeated below in its entirety.

> 6. A method for performing correctness checks, the method comprising the steps of:
>
> receiving a first set of instructions;
>
> ***generating an initial instruction schedule and a conditional instruction stream from the first set of instructions***, such that the initial instruction schedule is devoid of code sequences comprising correctness check

> functions and such that code sequences of the conditional instruction stream are associated with a corresponding set of one or more instructions in the initial instruction schedule;
>
> evaluating the initial instruction schedule to determine whether the initial instruction schedule includes spare instruction slots into which said code sequences associated with correctness check functions can be inserted into the initial instruction schedule such that a final instruction schedule would not require a longer run time than the initial instruction schedule; and
>
> inserting said code sequences from the conditional instruction stream and associated with the correctness check function into the spare instruction slots if enough spare instruction slots exist in the initial instruction schedule for accommodating said code sequences.

(Applicant's independent Claim 6 - *emphasis added.*)

The cited art of record fails to disclose, teach, or suggest at least the emphasized feature of pending claim 6 as shown above. Consequently, claim 6 is allowable.

Specifically, the systems disclosed in *Hooker* and *Austin* fail to disclose, teach, or suggest Applicant's claimed feature of "***generating an initial instruction schedule and a conditional instruction stream from the first set of instructions . . .***" Both *Hooker* and *Austin* are silent regarding generating an initial schedule and a conditional instruction stream from the first set of instructions.

For at least the reason that the proposed combination of *Hooker* and *Austin* fails to disclose, teach, or suggest this feature, claim 6 is allowable over the proposed combination and the rejection should be withdrawn.

Because independent claim 6 is allowable, dependent claim 7 is also allowable. *See In re Fine, supra.* Accordingly, Applicant respectfully requests that the rejection of claim 7 be withdrawn.

4. Claims 9 and 10

For convenience of analysis, independent claim 9 is repeated on the following page in its entirety.

> 9. A computer program for performing correctness checks, the computer program being embodied on a computer-readable medium, the computer program comprising:
>
> a first code segment configured to receive a set of instructions;
>
> ***a second code segment configured to generate an initial instruction schedule and a conditional instruction stream from the set of instructions***, such that the initial instruction schedule is devoid of code sequences comprising correctness check functions and such that the code sequences of the conditional instruction stream are associated with a corresponding set of one or more instructions in the initial instruction schedule;
>
> a third code segment configured to evaluate the initial instruction schedule to determine whether the initial instruction schedule includes spare instruction slots into which said code sequences associated with the correctness check function can be inserted into the initial instruction schedule such that a final instruction schedule would not require a longer run time than the initial instruction schedule; and
>
> a fourth code segment configured to insert code sequences associated with the correctness check function into the spare instruction slots when sufficient spare instruction slots exist in the initial instruction schedule to accommodate said code sequences.

(Applicant's independent Claim 9 - *emphasis added.*)

The cited art of record fails to disclose, teach, or suggest at least the emphasized element of pending claim 9 as shown above. Consequently, claim 9 is allowable.

Specifically, the systems disclosed in *Hooker* and *Austin* fail to disclose, teach, or suggest Applicant's claimed "***second code segment configured to generate an initial instruction schedule and a conditional instruction stream from the set of instructions . . .***" Both *Hooker* and *Austin* are silent regarding generating an initial schedule and a conditional instruction stream from the first set of instructions.

For at least the reason that the proposed combination of *Hooker* and *Austin* fails to disclose, teach, or suggest this element, claim 9 is allowable over the proposed combination and the rejection should be withdrawn.

Because independent claim 9 is allowable, dependent claim 10 is also allowable. *See In re Fine, supra.* Accordingly, Applicant respectfully requests that the rejection of claim 10 be withdrawn.

5. Claims 8 and 11

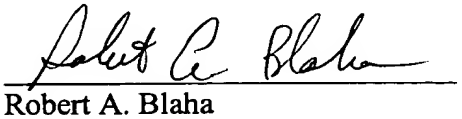
Claim 8 depends from allowable independent claim 6. Claim 11 depends from allowable independent claim 1. Claim 8 is allowable for at least the reason that the proposed combination of *Hooker*, *Austin*, and *Houldsworth* fails to disclose, teach, or suggest Applicant's claimed feature of generating an initial schedule and a conditional instruction stream from the first set of instructions. Claim 11 is allowable for at least the reason that the proposed combination of *Hooker*, *Austin*, and *Houldsworth* fails to disclose, teach, or suggest Applicant's claimed elements of "***logic configured to generate an initial instruction schedule and a conditional instruction stream from the first set of instructions . . .***," and "***logic configured to generate the final instruction schedule responsive to the initial instruction schedule, the conditional instruction stream, and the logic configured to evaluate***." Accordingly, Applicant respectfully requests that the rejection of claims 8 an 11 be withdrawn.

CONCLUSION

In summary, Applicant respectfully requests that all outstanding claim rejections be withdrawn. Applicants respectfully submit that all pending claims 1 - 15 are allowable over the cited art of reference and the present application is in condition for allowance. Accordingly, a Notice of Allowance is respectfully solicited. Should the Examiner have any comment regarding the Applicant's response or believe that a teleconference would expedite prosecution of the pending claims, Applicant requests that the Examiner telephone Applicant's undersigned attorney.

Respectfully submitted,

THOMAS, KAYDEN, HORSTEMEYER & RISLEY, L.L.P.

By: 

Robert A. Blaha
Registration No. 43,502
(770) 933-9500